

EXPRESS MAIL number: EF319723794US

Date of Deposit: April 4, 2001

I hereby certify that this paper is being deposited with the United States Postal Service "EXPRESS MAIL Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents; Washington, DC 20231.

Kelli J. Withrow

Typed name of person mailing paper or fee


Signature

=====

APPLICATION FOR UNITED STATES LETTERS PATENT

=====

Title: METHOD FOR SEGMENTING MULTI-RESOLUTION VIDEO OBJECTS

Inventors: Fatih M. Porikli
Yao Wang

1101010 00000000

Method for Segmenting Multi-Resolution Video Objects

Related Application

This application is related to U.S. Patent Application 09/xxx,xxx "*Method for*
5 *Measuring Compactness of Data and Signal Sets*" filed by Porikli on xxx. xx,
2001.

Field of the Invention

The present invention relates generally to video processing, and more particular to
video object segmentation.

Background of the Invention

Older video standards, such as ISO MPEG-1 and MPEG-2, are relatively low-level
specifications primarily dealing with the temporal and spatial compression of
entire videos.

Newer video coding standards, such as MPEG-4 and MPEG-7, see "Information
20 Technology -- Generic coding of audio/visual objects," ISO/IEC FDIS 14496-2
(MPEG4 Visual), Nov. 1998, allow arbitrary-shaped video objects to be encoded
and decoded as separate video object planes (VOP's). These emerging standards
are intended to enable multimedia applications, such as interactive video, where
natural and synthetic materials are integrated, and where access is universal. For
25 example, one might want to "cut-and-paste" a moving figure from one video to
another. In order to identify the figure, the video must first be "segmented." It is

possible to segment video objects under user control, i.e., semi-automatic, or unsupervised, i.e., fully automatically.

In the semi-automatic case, a user can provide a segmentation for the first frame of the video. The problem then becomes one of video object tracking. In the fully automatic case, the problem is to first identify the video object, then to track the object through time and space. Obviously, no user input is optimal.

With VOP's, each frame of a video is segmented into arbitrarily shaped image regions. Each VOP describes a video object in terms of, for example, shape, color, motion, and texture. The exact method of producing VOP's from the video is not defined by the above standards. It is assumed that "natural" objects are represented by shape information, in addition to the usual luminance and chrominance components. Shape data can be provided as a segmentation mask, or as a gray scale alpha plane to represent multiple overlaid video objects. Because video objects vary extensively with respect to low-level features, such as, optical flow, color, and intensity, VOP segmentation is a very difficult problem.

A number of segmentation methods are known. Region-based segmentation methods include mesh-based, motion model-based, and split-and-merge. Because these methods rely on spatial features, such as luminance, they may produce false object boundaries, and in some cases, foreground video objects may be merged into the background. More recently, morphological spatio-temporal segmentation has been used. There, information from both the spatial (luminance) and temporal (motion) domains are tracked using vectors. This complex method can erroneously assign a spatial region to a temporal region, and the method is difficult to apply to a video including more than one object.

Generally, unsupervised object segmentation methods can be grouped into three broad classes: (1) region-based methods that use a homogeneous color criterion, see M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second generation image coding," *Proc. IEEE*, no.73, pp.549-574, 1985, (2) object-based approaches that use a homogeneous motion criterion, and (3) object tracking.

Although color-based methods work well in some situations, for example, where the video is relatively simple, clean, and fits the model well, they lack generality and robustness. The main problem arises from the fact that a single video object can include multiple different colors.

Motion-oriented segmentation methods start with an assumption that a semantic video object has homogeneous motion, see B. Duc, P. Schtoeter, and J. Bigun, "Spatio-temporal robust motion estimation and segmentation," *Proc. 6th Int. Conf. Comput. Anall. Images and Patterns*, pp. 238-245, 1995. These methods either use boundary placement schemes, or region extraction schemes, see J. Wang and E. Adelson, "Representing moving images with layers," *IEEE Trans. Image Proc.*, no.3, 1994. Most of these methods are based on rough optical flow estimation, or unreliable spatio-temporal segmentation. As a result, these methods suffer from the inaccuracy of object boundaries.

The last class of methods for object segmentation uses tracking, see J. K. Aggarwal, L. S. Davis, and W. N. Martin, "Corresponding processes in dynamic scene analysis", *Proc. IEEE*, no.69, pp. 562-572, May 1981. However, tracking methods need user interaction, and their performance depends extensively on the initial segmentation. Most object extraction methods treat object segmentation as

an inter- or intra-frame processing problem with some additional parametric motion model assumptions or smoothing constraints, and disregard 3D aspect of the video data.

5 Therefore, there is a need for a fully automatic method for precisely segmenting any number of objects in a video into multiple levels of resolution. The method should use both motion and color features over time. The segmentation should happen in a reasonable amount of time, and not be dependent on an initial user segmentation, nor homogeneous motion constraints.

Summary of the Invention

The present invention provides a video object segmentation method that detects object boundaries precisely, without user assistance. A raw color or gray-scale video, or a processed video, e.g. with detected edges, successive-frame-difference, or texture score, is converted to a structure called a *video-volume*. Advanced 3D signal processing techniques are then applied to the volume.

10 The video is first filtered, the video volume is formed, and marker pixels are identified using, e.g., the color gradient of the pixel. A video volume is “grown” around each marker using color and texture distance criteria. Self-descriptors are assigned to volume, and mutual descriptors are assigned to pairs of similar volumes. These descriptors capture motion and spatial information of the volumes.

15 While applying and relaxing descriptor-based adaptive thresholds, similarity scores are determined for each possible pair-wise combination of volumes. The pair of volumes that gives the largest score is combined iteratively. In the combining

stage, volumes are classified and represented in a multi-resolution coarse-to-fine hierarchy of video objects.

More specifically, the method according to the invention segments a video
5 sequence of frames into video objects. Each frame is composed of pixels. Feature
vector are assigned to each pixel of the video. Next, selected pixels are identified
as marker pixels. Pixels adjacent to each marker pixel are assembled into a
corresponding a volume of pixels if the distance between the feature vector of the
marker pixel and the feature vector of the adjacent pixels is less than a first
predetermined threshold.

After all pixels have been assembled into volumes, a first score and self-descriptors
are assigned to each volume. At this point, each volume represents a segmented
video object.

The volumes are then sorted a high-to-low order according to the first scores, and
further processed in the high-to-low order.

Second scores, dependent on the descriptors of pairs of adjacent volumes are
20 determined. The volumes are iteratively combined if the second score passes a
second threshold to generate a video object in a resolution video object tree that
completes when the combined volume or video object is the entire video.

Brief Description of the Drawings

Figure 1 is a block diagram of video volumes according to the invention;

- 5 Figure 2 is a flow diagram a video segmentation method according to the invention;

Figure 3 is a block diagram of video volume self descriptors;

- 10 Figure 4 is a block diagram of video volume mutual descriptors;

Figure 5 is a flow diagram of a volume combining step of the method of Figure 2;
and

- 15 Figure 6 is a multi-resolution video object tree produced by the method of Figure 2;

Figure 7 is a block diagram of video volume self and mutual descriptors;

- 20 Figure 8 is a diagram of fast 2-D median filtering done within 3x3 blocks; and

Figure 9 is a flow diagram of small volume merging according to the invention.

Detailed Description of the Preferred Embodiment

Color, Edge, and Difference-Map Video Volumes

5 As shown in Figure 1, our invention arranges 100 of a video 101, e.g., sequences of frames 1-N 102, into three-dimensional (3D) three-dimensional data structures V i.e., *video volumes* 111-113. The color, edge, and difference-map volumes 111-113 have identical spatial (x,y) 104 and time (t) 104 axes. Then, we apply advanced 3D processing techniques 200 to the video volumes 111-113 to robustly segment video objects into a multi-resolution hierarchy 600.

A video volume $V(x,y,t)$ is defined as for a spatial-temporal collocated scene of the video 101 between two scene cuts 121-122. For a streaming video, the video volumes 111-113 can be generated for a certain number (N) of frames to allow overlap. This ensures object consistency within the volumes.

In case of moderate object motion, a portion of a video object in one frame intersects its projections on adjacent frames. Thus, object portions in the color-based video volumes have continuous silhouettes along the time axis t 104.

Object Segmentation

As shown in Figure 2, the video volumes 111-113 can be formed from raw color data (R,G,B or Y,U,V) 201 in the frames 102, or from processed frames, i.e., including edges 237, texture scores 238, successive frame differences 239, etc. 202, hereinafter "features."

The frames 102 are indexed ($1-N$), and a 3×3 spatial-domain 2D median filter 210 is applied to the frames 102 in order to remove intensity singularities, without disturbing edge formation. We utilize a 2D median filter that exploits 2D coherence, and accelerates the computationally intensive filtering step.

5

We also determine two horizontally adjacent medians to reduce necessary comparisons described below. We prefer not to use a 3D median filter so that motion is preserved.

Fast Median Filtering

Figure 8 shows a fast median filter 211 that exploits 2-D coherence. Two horizontally adjacent medians are determined in a single step by reducing the necessary comparisons to find median (5^{th} in a 9 elements list) within a 3×3 window from 30 to 9.5. First, the slices c 803 and d 804 are sorted as slices a 801 and b 802 of a previous step with six comparison. Sorting is done by a set of nested “if” conditions. Then, slices b 802 and c 803 are merged in a slice bc using five comparison. Slices a 801 and bc are merged to determine compute a median for p/r with four comparison. Finally, slices d 804 and bc are merged to determine a median for q/r with four comparison. In the worst case, this process takes a total of 19 comparisons.

20

To prevent over-segmentation in a volume growing step 240, described below, a 2D smoothing filter 220 is applied to the median filtered frames. We prefer a 2D Gaussian filter with a 5×5 spatial window. Again, a 3D filter is not used to preserve motion.

25

Marker Identification

The volume growing process 240 connects the pixels of $V(x,y,t)$ such that color and texture distributions of the connected pixels are uniform. Such grouped pixels, called *volumes*, are expanded from some seed pixels, called markers. The marker pixels m_i can be selected from the pixels of the entire volume of pixels in three ways.

Uniformly Distributed Markers

The video volume V is divided into identical smaller volumes and the centers of the smaller volumes are selected as markers.

Minimum Gradient Markers with Fixed Neighborhood

A set S initially contains all possible spatio-temporal pixels of the volume V . For each pixel, a 3-D gradient

$$\nabla V = \partial V / \partial x + \partial V / \partial y + \partial V / \partial t$$

is computed from the color components. Then, the pixel with the minimum gradient is identified 230 as a marker pixel. Pixels in a predetermined neighborhood around the marker are removed from the set S . The next minimum in the remaining set is chosen, and the identification process is repeated until no pixel remains in the set S .

Minimum Gradient Markers with Volume Growing

The minimum m_i is chosen as above. Instead of removing the pixels adjacent to the marker pixel, a volume P_i is assembled, according to

$m_i = \arg \min \nabla V(x, y, t)_{r,g,b}$, $S = V - \bigcup_{j=1}^i P_j$, until all the pixels of the volume are removed from the set S .

The next minimum in the remaining set is chosen, and identification process
5 repeated until no pixel remains in the set.

Assembling Volumes

The volumes P_i $i=1,..,M$ are assembled 240 around the markers m_i pixels according to the features, e.g., color, texture, etc., similarity criteria. We assign a feature vector $\mathbf{m}(x,y,t)$ to each pixel in the video volume $V(x,y,t)$. Minimally, the feature vector specifies the color components 201 of the pixel. Optionally, the feature vector can also include other data 202, such as texture scores 238 obtained by applying Gabor filters. If we only use the color feature, the feature vector \mathbf{m}_i for a marker pixel m_i is

$$\mathbf{m}_i = [R(m_i), G(m_i), B(m_i)]^T$$

Distances d between feature vectors \mathbf{n}_j of adjacent pixels and the feature vector \mathbf{m}_i of marker m_i are measured as

$$d = \|\mathbf{m}_i - \mathbf{n}_j\|.$$

If the distance d is smaller than a predetermined threshold λ , then the adjacent pixel is included in the volume P_i , and the adjacent pixel is set as an active surface pixel for the volume P_i .

Next, the feature vector for the marker pixel is updated as

$$d \leq t \Rightarrow \begin{cases} m_i^{k+1} = (N_i m_i^k + n_j) / (N_i + 1) \\ N_i = N_i + 1 \end{cases}$$

In the next iteration, the adjacent pixels of the active surface pixels are compared.

- 5 This operation is repeated until all pixels in the video volume are processed. The above process assembles adjacent pixels with similar feature vectors as the marker pixel into the same volume. The location of each assembled volume is designated by the location of its marker pixel.

Small Volume Merging

Volumes that are less than a minimum size are merged 250 with adjacent volumes as shown in Figure 9. For example, volumes less than 0.001 of the volume V , i.e., the entire video. To accelerate the searching process, the merging 250 is performed in a hierarchical manner by starting with the smallest volume, and ending with the largest volume that does not satisfy the minimum size requirement. The smallest volume that does not satisfy the requirement is chosen 251. All the pixels of the smallest volume are unmarked 252. Then, for each unmarked pixel, a closest volume P_c located 254; and the pixel is included 255 in that volume. Steps 253-255 are repeated for all pixels, and all small volumes.

Volume Descriptors

Next as shown in Figure 3, 4, and 7, we assign a set of self descriptors $F(P_i)$ 300 and a set of mutual descriptors $F(P_i, P_j)$ 400 to each volume P_i . These descriptors

are used to identify the motion (trajectory) 701, shape or spatial and color 702 characteristics of the volumes, as well as the mutual correlation between any pair of volumes P_i , P_j . The descriptors 300 and 400 are assigned 260 as follows.

- 5 A trajectory T_i is determined for each volume P_i by averaging the vertical and horizontal coordinates of pixels inside the volume, frame-wise, as described below. Instead of averaging, other center-of-mass definitions can also be used. The self-descriptor $F_1(P_i)$ 300 is a vector that includes the color averages of the pixels in the volume. The color mean (average) 301 includes red, green, blue components for a RGB image, and hue, saturation, and intensity for a YUV image.

Self-Descriptors

The color of a pixel p_k is denoted as $R(p_k)$, e.g., for the red color component. Then, $F_1(P_i)$ stands for the mean of the red color component. $F_2(P_i)$ 302 represents the number of pixels in the volume. $F_3(P_i)$ 303 is the number of pixels on the surface of the volume. A first compactness 304 is defined as a ratio of volume to squared surface is $F_4(P_i)$. A second compactness descriptor $F_5(P_i)$ 305 is defined by using maxcord instead of surface.

For further detail on the preferred compactness measure, please see U.S. Patent Application 09/xxx,xxx “*Method for Measuring Compactness of Data and Signal Sets*” filed by Porikli on xxx. xx, 2001, incorporated herein by reference.

- 25 Maxcord is the length of the longest cord that can fit in the volume. $F_6(P_i)$ and $F_7(P_i)$ 306-307 describe the trajectory of a volume in horizontal direction and vertical direction, respectively, for the sequence of frames. $F_8(P_i)$ 307 is the total

length (route length) of the trajectory. $F_9(P_i)$ 309 and $F_{10}(P_i)$ 310 are averaged coordinates of the volume's pixels.

Mutual Descriptors

5

Mutual descriptors $F(P_i, P_j)$ 400 express the spatial, shape, motion, color relation between volumes. $F_{11}(P_i, P_j)$ 411 is the averaged distance between the trajectories of volumes P_i, P_j by summing the distance of trajectories at each frame where both volumes exist. The variance of trajectory distance is $F_{12}(P_i, P_j)$ 412, and its maximum is $F_{13}(P_i, P_j)$ 413. Average change in distance $F_{14}(P_i, P_j)$ 414 stands for the accumulated distance change of trajectories between frames. Direction of a volume is the vector pointing from the volume's center-of-mass in the last frame to the center-of-mass of the volume in the frame where it existed. Direction difference $F_{15}(P_i, P_j)$ 415 is the distance of such two vectors associated with the volumes P_i, P_j . $F_{16}(P_i, P_j)$ 416 expresses the compactness of the mutual volume in terms of the average of their separate compactness scores. $F_{17}(P_i, P_j)$, $F_{18}(P_i, P_j)$ 417-418 are mutual volume and surface. $F_{19}(P_i, P_j)$ 419 is the color difference, and $F_{20}(P_i, P_j)$ 420 is the number of frames both volume coexists.

20 Volume Combining

The volumes are combined with respect to their descriptors in a clustering step 500 in order to segment the video into multi-resolution video objects. For each volume P_i , we determine a trajectory $T_i(t)=(x_t, y_t)$ by taking the spatial averages of the
25 volume's pixels on a per frame basis.

$$T_i(t) = (x_t, y_t) = \frac{1}{N_i} \left(\sum_{p \in P_i, t} x, \sum_{p \in P_i, t} y \right).$$

Then, the distance $\Delta d_{ij}(t)$ between the trajectories of two volumes P_i and P_j , at time t is

$$\Delta d_{ij}(t) = |T_i(t) - T_j(t)|.$$

- 5 The motion information such as vertical and horizontal motion, route length, mean and variance of distance, direction difference, and average change in the distance are extracted from the trajectories. Therefore, without estimating motion by optical flow, parametric models or extensive search-based matching methods as in the prior art, our method uses the motion information efficiently.

As shown in Figure 5, the clustering step 500 produces segmented video objects 600 by iteratively merging volumes having substantially similar pair of volumes descriptors 300, 400. First, the descriptors 300 of the volumes P_i are scored 510 with respect to weighted averages of their sizes, compactness and existence values. The scored volumes are next sorted in a high-to-low order.

Starting with the first volume P_i in the sorted list, each volume is processed as follows, until a single volume remains 599.

- 20 During the merging, the descriptors of the current volume 511 are compared 520 to the descriptors of its adjacent volumes. If their descriptors pass 530 a set of adaptively constraint thresholds τ_k $k=10, \dots, 20$, determine 540 a similarity score

$$S(P_i, P_j) = \omega_1 F_{11}(P_i, P_j) + \omega_2 F_{12}(P_i, P_j) + \omega_3 F_{14}(P_i, P_j) + \omega_4 F_{15}(P_i, P_j) + \omega_5 F_{16}(P_i, P_j).$$

- The size of the threshold defines resolution of the video objects that will be
25 segmented.

If the descriptors of the adjacent volumes pass 531 the constraint tests 530, then the adjacent volume with the largest similarity score is selected 550 and combined 560 with the current volume 511, and the thresholding and combining continues with the next volume 511 in the sorted list. If no volumes are combined during an
5 iteration, the thresholds are relaxed 570 using

$$\tau(F_i)^{k+1} = \tau(F_i)^k \pm \alpha \frac{1}{|\max F_i - \min F_i|}.$$

Combining continues, until the similarity scores exceed the initial scores by a substantial margin, or no volumes remain.

Multi-Resolution Object Tree

As shown in Figure 6, the volumes during the iterations of merging can be represented as nodes in a multi-resolution object tree having N levels 601, one level for each iteration. Level 1 represents the entire video, level 2 segments the video into moving video objects 621 and background 622. The background volume has slow motion over time, is consistent spatially, and relatively large 623 compared to the moving video objects 621. Lower in the multi-resolution tree, video objects with consistent motion video volumes are segmented 631. At the
20 bottom level of the tree, i.e., level N , the video objects with uniform color, uniform texture, uniform shape, and spatial connectivity 641 are correctly segmented using the video volumes according to our invention.

Our method for segmenting video objects is robust, even when the motion of the
25 video objects in the video is large. Moreover, our method is considerably faster than methods that rely on computing dense optical flows. The method enables a

5

10